www.bytescout.com

How to PDF text search API in PowerShell using ByteScout Cloud API Server

This code in PowerShell shows how to PDF text search API with this how to tutorial

These source code samples are assembled by their programming language and functions they apply. ByteScout Cloud API Server: API server that is ready to use and can be installed and deployed in less than 30 minutes on your own Windows server or server in a cloud. It can save data and files on your local server-based file storage or in Amazon AWS S3 storage. Data is processed solely on the API server and is powered by ByteScout engine, no cloud services or Internet connection is required for data processing.. It can PDF text search API in PowerShell.

The SDK samples given below describe how to quickly make your application do PDF text search API in PowerShell with the help of ByteScout Cloud API Server. Just copy and paste the code into your PowerShell application's code and follow the instructions. Complete and detailed tutorials and documentation are available along with installed ByteScout Cloud API Server if you'd like to learn more about the topic and the details of the API.

You can download free trial version of ByteScout Cloud API Server from our website to see and try many others source code samples for PowerShell.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Cloud API Server

Explore API Documentation

Get Free Training for ByteScout Cloud API Server

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

PDFTextSearchFromUrlAsynchronously.ps1

```
# Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://lo
# If it's not then please replace this with with your hosting url.

# Direct URL of PDF file to get information
$SourceFileURL = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pd

# Comma-separated list of page indices (or ranges) to process. Leave empty for all pag
$Pages = ""

# PDF document password. Leave empty for unprotected documents.
$Password = ""

# Search string.
$SearchString = '\d{1,}\.\d\d' #Regular expression to find numbers like '100.00'

# Enable regular expressions (Regex)
$RegexSearch = 'True'

# (!) Make asynchronous job
$Async = $true

# Prepare URL for PDF text search API call.
$query = "https://localhost/pdf/find?password=$($Password)&pages=$($Pages)&url=$($Sour
$query = [System.Uri]::EscapeUriString($query)

try {
    # Execute request
    $jsonResponse = Invoke-RestMethod -Method Get  -Uri $query

    if ($jsonResponse.error -eq $false) {
        # Asynchronous job ID
        $jobId = $jsonResponse.jobId

        # URL of generated JSON file with search result that will available after the
        $resultFileUrl = $jsonResponse.url

        # Check the job status in a loop.
        # If you don't want to pause the main thread you can rework the code
        # to use a separate thread for the status checking and completion.
        do {
            $statusCheckUrl = "https://localhost/job/check?jobid=" + $jobId
            $jsonStatus = Invoke-RestMethod -Method Get  -Uri $statusCheckUrl

            # Display timestamp and status (for demo purposes)
            Write-Host "$(Get-date): $($jsonStatus.status)"

            if ($jsonStatus.status -eq "success") {
                # Get JSON for search result
                $jsonSearchResult = Invoke-RestMethod -Method Get  -Uri $resultFileUrl

                # Display found result in console
                foreach ($item in $jsonSearchResult)
                {
```

```
                    Write-Host "Found text $($item.text) at coordinates $($item.left),
                }
                break
            }
            elseif ($jsonStatus.status -eq "working") {
                # Pause for a few seconds
                Start-Sleep -Seconds 3
            }
            else {
                Write-Host $jsonStatus.status
                break
            }
        }
        while ($true)
    }
    else {
        # Display service reported error
        Write-Host $jsonResponse.message
    }
}
catch {
    # Display request error
    Write-Host $_.Exception
}
```

run.bat

```
@echo off

powershell -NoProfile -ExecutionPolicy Bypass -Command "& .\PDFTextSearchFromUrlAsynch
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Cloud API Server Home Page
Explore ByteScout Cloud API Server Documentation
Explore Samples
Sign Up for ByteScout Cloud API Server Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co